Prime Number & Sieve

Bruce

Prime Number

- A prime number is an integer $p > 1$ which is only divisible by 1 and itself. In other words, if p is a prime number, then $p = a \cdot b$ for integers $a \leq b$ implies that $a = 1$ and $b = p$.
- Every integer can be expressed in only one way as the product of primes.

  105 = 3 * 5 * 7

  The unique set of numbers multiplying to *n is called the prime factorization of n.*

Find Primes

- Easiest way: repeated division (O(n))
  - Start from the smallest candidate divisor, and then try all possible divisors up from there.

```
bool Brute_Force(int n)
{
   for (int i=2;i<=n-1;i++)
            if (n%i==0) return false;
            return true;
}
```

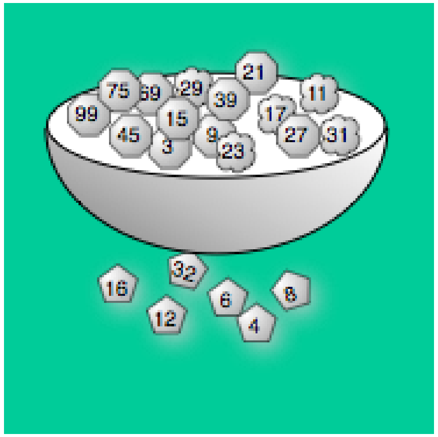
- Improved method (O(sqrt(n)))

```
boolean isPrime(int n)
{
    for (int i=2; i*i<=n; i++)
            if (n % i==0)  return false;
            return true;
}
```

Sieve of Eratosthenes

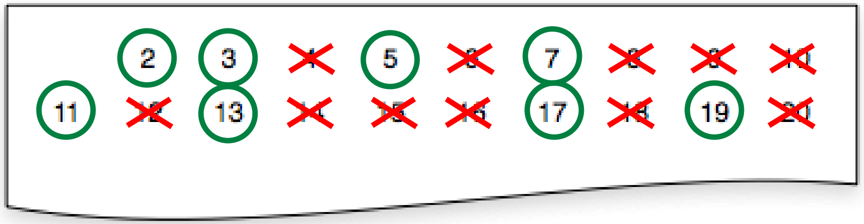
- Sieve of Eratosthenes is a simple, ancient algorithm for finding all prime numbers up to any given limit.
- Iteratively marking as composite the multiples of each prime, starting with the multiples of 2. The multiples of a given prime are generated starting from that prime, as a sequence of numbers with the same difference, equal to that prime, between consecutive numbers.

The Sieve



- The basic idea is simple:

```
make a list of numbers, starting with 2
repeat:
          the first number in the list is prime
          cross off multiples of the most recent prime
```



See “Sieve of Eratosthenes” at Wikipedia

```
#define MAX 10007
bool isprime[MAX];
void TheSieveofEratosthees() {
        int i,j;
        for (i=2;i<MAX;i++)
                    isprime[i]=1;
        for (i=2;i<MAX;i++)
        {
             if (isprime[i])
                    for (j=i+i;j<MAX;j+=i)
                                 isprime[j]=0;
        }
}
```

Using O(nlglgn) time to get all primes and then using O(1) time to verify prime number

Prime Factorization

- "Prime Factorization" is finding **which prime numbers** multiply together to make the original number.

```
int x = in.nextInt(), y = (int)Math.sqrt(x);
for(int i=2; i<=y; i++){
     while(x % i == 0){
               System.out.print(i + " "); x /= i;
      }
 }
if(x != 1) System.out.println(x);
else System.out.println();
```

Count Primes

- How many primes are there?
- Infinite – by Euclid’s proof
- Prime Number Theorem
  - if a random integer is selected in the range of zero to some large integer N, the probability that the selected integer is prime is about 1 / ln(N), where ln(N) is the natural logarithm of N.
  - N = 10^3 about one in seven numbers is prime,
    N = 10^{10} about one in 23 numbers is prime (where ln(10^3)= 6.90775528. and ln(10^{10})=23.0258509)